MATCHED TEXTURE FILTER DESIGN FOR RENDERING

MULTI-RATE DATA SAMPLES

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001]   This application claims the benefit of Provisional Patent Application Serial No. 60/204,085, filed on May 12, 2000, entitled "A Matched Texture Filter Design for Rendering Multi-Rate Data Samples," and is related to a commonly-assigned U.S. Patent Application Serial No._____ filed on May 14, 2001, entitled "Selective Super-Sampling/Adaptive Anti-Aliasing or Complex 3D Data", the subject matter of which is herein incorporated by reference.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002]   The present invention relates generally to graphics systems, and more particularly to texture filtering for 3D graphics with screen pixel digitization at variable sampling rates.

Description of the Background Art

[0003]   A texture is a digital image, typically rectangular in shape, having a (u, v) coordinate space. The smallest addressable unit of a texture is a texel, which is located at a particular coordinate in the (u, v) coordinate system. In a texture mapping operation, a texture is mapped to the surface of a graphical model as the model is rendered to create a destination image. In the destination image, pixels are located at particular coordinates in the (x, y) coordinate system.

[0004]   In texture mapping, data is sampled from the texture to compute an intensity value for a pixel in the destination image. The sampled texels for a given pixel are typically identified by mapping a point in the destination image into the texture.

Texels neighboring the mapped point in the texture are then sampled, weighted, and summed to compute the intensity value for a pixel. The process is then repeated for additional pixels in the destination image.

[0005] Various texture reconstruction filtering techniques may be used to reduce blocky appearance and aliasing noise in textures while applying perspective effects to produce a smoother and more realistic destination image. Such texture filtering techniques include box filtering (point sampling or any box filter), bilinear filtering, trilinear filtering, and anisotropic filtering.

[0006] Filtering with the box filter (point sampling) technique is the most basic method for texture mapping and determining the attributes, such as color, of a pixel from the texture maps. Point sampling uses the nearest texel value mapped to a particular pixel to determine the pixel attribute. For example, assume that a given pixel location at coordinates (x, y) is mapped to particular texture coordinates (u, v). The texture coordinates (u, v) may include fractional values. However, in point sampling, the texture coordinate assigned to the pixel will be the nearest integer value. Thus, if a given pixel is mapped to texture coordinates $(u, v) = (1.2, 1.9)$, the point sampling technique will assign a texel with texture coordinates (u', v') to the given pixel with the texture coordinates (u', v') being based on the nearest integer value: $(u', v') = (1, 2)$.

[0007] Bilinear filtering involves interpolating the attribute values of four surrounding discrete texels to calculate the attribute value of a new texel which is then assigned to a particular pixel. The process is then repeated for each pixel forming the object being textured. The bilinear filtering technique provides reasonably realistic images and improved image quality over the point sampling filtering technique.

[0008] Trilinear filtering involves the use of multiple texture maps or MIP maps at various levels of depth. Bilinear filtering is preformed in each MIP map level, resulting in a new texel for each MIP map level. The actual attribute value for a pixel is then determined by performing another interpolation step between the two new texels associated with two particular MIP map levels, resulting in an interpolation process in three dimensions.

[0009] Anisotropic filtering involves taking samples of texels in the texture with the samples chosen according to the shape of the pixel "footprint" in the texture

2

space. Thus, the shape of the region of the texels that are sampled varies as circumstances dictate.

[0010] These various texture-filtering algorithms do not address the problem of variable sampling rate or other non-standard graphics processes. When sections of a scene are sampled at varying sampling rates, it is desirable to appropriately modify the characteristics of the texture filter. Otherwise, the effective filter characteristics will not be consistent over the entire scene, resulting in undesirable artifacts.

[0011] Further, when a scene is super-sampled, a convolution is effectively performed for the post-filter and the texture reconstruction filter. The characteristics of the resulting higher-order filter typically matches the characteristics of a filter used for coarse sampling. Without this matching of filter characteristics, banding will occur in the image due to the additional filtering applied to aggregate dense samples resulting in different passbands in the areas with the different sampling rates.

[0012] Additionally, in conventional systems, the texture-filtering model is typically pre-specified in each particular application. Conventional systems do not provide a texture filter that varies the filter characteristics in response to, for example, the rate of sampling and the extent of texture warping. Such deficiencies of conventional systems lead to images that are generated with undesirable artifacts. Therefore, there is a need for a matched texture filter design for varying sample rates.

SUMMARY OF THE INVENTION

[0013] The present invention provides in various embodiments a texture filter that takes into account the sampling rate and the non-linear mapping between screen coordinates and texture coordinates. In particular, these embodiments of the present invention provide a matched texture filter that continuously changes the pass band – stop band characteristics based upon the rate of sampling of polygon data and the extent of perspective warping introduced by the non-linear screen-to-texture mapping function. The matched texture filter is capable of changing so that after post-filtering, the resulting filter characteristic effectively matches the characteristics of a filter used for coarse sampling.

[0014] In one embodiment of the present invention, an apparatus is provided for texture filtering with characteristics that continuously vary between a bilinear filter and a box filter. The variations may be based upon, for example, the sampling rate of polygon data and the degree of texture warping. The texture filter behaves like a bilinear filter when the sampling rate per pixel is at the same rate as the desired screen resolution. The texture filter may have the characteristic of a combination of a bilinear filter and a box filter when the sampling rate increases per pixel. The characteristics of the matched texture filter can also be dependent upon the amount of "stretching" or "squeezing" of the texture area when polygon data is super-sampled. A texture area is "stretched out" when a small texture area (e.g., one texel) fills a big screen area. A texture area is "squeezed" when multiple texels are imposed on a small screen area.

[0015] In a second embodiment of the present invention, a texture filter is provided that behaves like a trilinear filter when the sampling rate per pixel is at the same rate as the desired screen resolution. The texture filter is a combination of a linear filter (between MIP levels) and a box filter when the sampling rate increases per pixel. As in the first embodiment, the characteristics of the matched texture filter in the second embodiment can also be dependent upon the amount of "stretching" or "squeezing" of the texture area when the polygon data is super-sampled.

[0016] In a further embodiment, the variation of the filter characteristics between the various filter modes (e.g., bilinear filter, trilinear filter, point sample filter, etc.) is preferably continuous. By continuously varying texture filters between two

modes, banding in images and artifacts normally visible during animation are eliminated.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017]   FIG. 1 is a block diagram of a system for performing the texture filtering operations in accordance with an embodiment of the present invention;

[0018]   FIG. 2 is a graph illustrating the spatial characteristics of the point sample filter, the bilinear filter and the matched texture filter in accordance with an embodiment of the present invention;

[0019]   FIG. 3 is a diagram illustrating an example of multi-rate polygon data sampling;

[0020]   FIG. 4 is a diagram illustrating an example of a texel mapped to a large screen area;

[0021]   FIG. 5 is a diagram illustrating an example of multiple texels mapped to a smaller screen area;

[0022]   FIG. 6 is a block diagram illustrating an internal edge problem; and

[0023]   FIG. 7 is a block diagram illustrating the details of the matched texture filter in accordance with an embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

[0024] FIG. 1 is a block diagram of an exemplary system 100 for performing the texture filtering operations of the present invention. The system 100 is typically implemented in a graphics processing system within a computer or similar processing device. The system 100 includes a primitive processor 102, a multi-rate data sampler 104, a matched texture filter 106, RAM 108 for storing textures, a pixel processor 110 and a rendering buffer 112.

[0025] The primitive processor 102, which may be a triangle processor, typically receives the three-dimensional geometry elements (e.g., triangles or other graphic primitives) and processes the data describing the size, shape, position, and other relative characteristics of the graphics primitives. In some cases, the primitive processor 102 is also capable of generating edge functions of the primitives. These primitives may be defined in three-dimensions using Euclidian coordinates or in four-dimensions using homogenous coordinates, and subsequently, projected onto a two-dimensional plane by a known algorithm.

[0026] The multi-rate data sampler 104 selects a finite set of values from the polygon data received by the primitive processor 102. The sampling of the polygon data may occur at different resolutions. For example, interior portions of a polygon may be sampled at a rate which is required to generate the destination screen resolution, while the detected edges of a polygon may be super-sampled at a higher resolution.

[0027] The matched texture filter 106 performs one of the filtering techniques (e.g., bilinear filtering, trilinear filtering, box filtering, a combination of bilinear and box filtering, and/or a combination of linear filtering between MIP levels) to calculate the color value (or other attribute) of a new texel which is then assigned to a particular pixel. The matched texture filter 106 may generate the filtered textured pixel values based on data received from the RAM 108. In a preferred embodiment, the matched texture filter 106 continuously changes its filter pass band – stop band characteristics based upon the rate of sampling and the extent of perspective warping introduced by the non-linear texture-to-screen mapping function. Additionally, the matched texture

filter 106 may be used in various types of applications such as in rendering multi-rate data samples (polygon data sampling at different resolutions).

[0028] The pixel processor 110 performs rendering operations. The rendering operations may be altered in a selected manner to generate various effects such as simulated light sources and shadows. Finally, the rendering buffer 112 stores images which may be displayed in a display device or used to render another image.

[0029] FIG. 2 is a graph illustrating spatial characteristics of a box filter, a bilinear filter, and a matched texture filter. The equation used to compute the coefficients β_u, β_v for each texture coordinates u, v is given by:

$$\beta_u = \min\{\delta_u \cdot (n-1)/n, 1.0\} \qquad (1)$$

$$\beta_v = \min\{\delta_v \cdot (n-1)/n, 1.0\} \qquad (2)$$

wherein β_u, β_v are coefficients that determine the blending factor α_u, α_v (for the u and v texture coordinate directions, respectively) for texture interpolation.

[0030] The term n is the (linear) sampling rate of the polygon data. The sampling rate n determines the characteristic of the matched texture filter 106 (FIG. 1). When $n = 1$, polygon data are sampled at a rate which is required to generate the destination screen resolution. For this condition, the matched texture filter 106 behaves as a pre-determined default filter (i.e., the matched texture filter 106 may behave like a bilinear or trilinear filter).

[0031] However, when $n > 1$, polygon data are sampled at a finer resolution. For example, polygon edges may be selectively super-sampled at higher resolution values as described in the commonly-assigned U.S. Patent Application entitled "Selective Super-Sampling/Adaptive Anti-Aliasing or Complex 3D Data". The interior of a primitive may be sampled at the resolution which is required during a scan conversion stage, while the primitive edge may be super-sampled at a higher resolution than that required during the scan conversion stage. For this condition, the matched texture filter 106 may be a combination of the pre-determined default filter (e.g., bilinear filter) and a low order filter such as a box filter, as illustrated by matched filter graph 206.

[0032] Alternatively, the matched texture filter 106, when $n > 1$, may embody characteristics of a linear filter (between MIP levels) and a combination of a bilinear and

8

box filter. Thus, the matched texture filter 106 is capable of morphing between different filters, such as a bilinear filter and a combination bilinear and low order filter. In one embodiment, the matched texture filter morphs in a continuous manner. Since, the matched texture filter 106 behaves with the characteristics of a combination of the predetermined default mode and a point sample filter when $n > 1$, an improved texture image is advantageously generated.

[0033] When $\beta_u = 0$, $\beta_v = 0$, then the matched texture filter 106 behaves as a bilinear filter and will have the characteristics of bilinear filter graph 204. When $\beta_u = 1$, $\beta_v = 1$ in equations (1) and (2), then the matched texture filter 106 behaves like a box reconstruction filter (effectively point sampling) and will have the characteristics of box filter graph 202. Finally, when $0 < \beta_u < 1, 0 < \beta_v < 1$, the matched texture filter 106 behaves with the characteristics of both a bilinear and point sample filter. Thus, the matched texture filter 106 has characteristics which morph between a bilinear and a box filter. The matching filter graph 206 is an exemplary illustration of the characteristics of the matched texture filter 106 when $0 < \beta_u < 1, 0 < \beta_v < 1$ is satisfied (i.e., $n > 1$ in equations (1) and (2)). As shown in FIG. 2, the matched filter characteristic (see graph 206) is a combination of characteristics of the bilinear filter (see graph 202) and the box filter (see graph 204). As n approaches infinity, the value β approaches a value of 1. It should be noted that β_u, β_v in equations (1) and (2) are restricted to values between 0 and 1. This restriction function may be implemented by, for example, software code.

[0034] The terms δ_u, δ_v represent the amount of "stretching" of the texture area (when a small texture area is mapped to a large screen area), or the amount of "squeezing" of a texture area (when a large texture area is mapped to a smaller screen area). Typically, the values of δ_u, δ_v are obtained, for example, from polygon data per sample. As shown in equations (1) and (2), the values of β_u, β_v are dependent on the values of δ_u, δ_v. Thus, the values of δ_u, δ_v determine the blending factors α_u, α_v for texture interpolation when the polygon data are super-sampled (i.e., when $n > 1$).

[0035] A conventional technique may be used to determine the amount of stretching. Assuming the texture mapping is defined by the (inverse) functions

9

$$u = U(x, y) \tag{3}$$

$$v = V(x, y) \tag{4}$$

where U and V are usually some projective (first order rational) functions computed during the primitive setup. A Jacobian is computed as follows:

$$J = \begin{pmatrix} \dfrac{\partial U(x,y)}{\partial x} & \dfrac{\partial V(x,y)}{\partial x} \\ \dfrac{\partial U(x,y)}{\partial y} & \dfrac{\partial V(x,y)}{\partial y} \end{pmatrix} \tag{5}$$

and then,

$$d_u = J^{-1}\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad d_v = J^{-1}\begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{6}$$

Thus, we calculate the δ_u, δ_v as:

$$\delta_u = \min\left\{\frac{1}{|d_u|}, 1\right\}, \quad \delta_v = \min\left\{\frac{1}{|d_v|}, 1\right\} \tag{7}$$

[0036] An example for determining the texel blending factors α_u is given below. FIG. 2 shows the graph for the term $(1 - \alpha)$ as a function of *uFraction* or *vFraction*. Given the fractional part (*uFraction*) of the texture coordinate u, the texel blending factor α_u is computed as follows (note that the algorithm may take on a slightly different form dependent on whether β takes on a full- or half-width value as shown in FIG. 2).

> *uFraction* = u − **floor**(u),
>
> If (*uFraction* $\leq \beta_u /2$)
>
> > $\alpha_u = 0.0$
>
> else if (*uFraction* $> (1.0 - \beta_u /2)$)
>
> > $\alpha_u = 1.0$
>
> else
>
> > $\alpha_u = (uFraction - \beta_u /2) / (1.0 - \beta_u)$.

[0037] When $\alpha_u = 0$, the value of $(1 - \alpha_u)$ is represented by segment 210 with respect to the u texture coordinate direction. When $\alpha_u = 1$, the value of $(1 - \alpha_u)$ is represented by segment 212 with respect to the u texture coordinate direction. Finally.

10

When $\alpha_u = (uFraction - \beta_u/2)/(1 - \beta_u)$, the value of $(1 - \alpha_u)$ is represented by segment 214 with respect to the u texture coordinate direction. Similarily,

$$vFraction = v - \mathbf{floor}(v),$$

If $(vFraction \leq \beta_v/2)$

$$\alpha_v = 0.0$$

else if $(vFraction > (1.0 - \beta_v/2))$

$$\alpha_v = 1.0$$

else

$$\alpha_v = (vFraction - \beta_v/2)/(1.0 - \beta_v).$$

[0038] FIG. 3 is a diagram 300 illustrating an example of multi-rate polygon data sampling. As shown in FIG. 3, if polygon data 302, 304, 306, 308 and 310 are each sampled at a rate required to generate the destination screen resolution, then $n = 1$ and the matched texture filter 106 (FIG. 1) behaves as a pre-determined default filter. However, if polygon data 320 – 344 are each super-sampled at a finer resolution, then n > 1 and the matched texture filter 106 is a combination of the pre-determined default filter and a low order filter.

[0039] FIG. 4 is an exemplary diagram illustrating the mapping of a small textual area (texel 400) to a large screen area 406. As shown, the small texture area (texel) 400 of texture 402 is mapped to the larger sized screen area 406 having a size greater than one pixel (such as pixels 408, 410 and 412). This mapping results in the stretching of the texel 400.

[0040] FIG. 5 is an exemplary diagram illustrating the mapping of multiple texels 502a-d of a texture 504 to a smaller screen area 506 having a size less than one pixel 508. This mapping results in the squeezing of the multiple texels 502a-d.

[0041] It should be noted that when super-sampling of polygon data is not performed (i.e., $n = 1$), β_u, β_v will have values of zero in equations (1) and (2), and the matched texture filter 106 (FIG. 1) will operate in the predetermined default mode (e.g., bilinear or trilinear filtering). Alternatively, when super-sampling of polygon data is performed (i.e., n > 1), the amount of texture "stretching" or "squeezing", as represented in the δ_u, δ_v values of equations (1) and (2) will partly determine the β_u,

β_v values. The β_u, β_v values, in turn, determine the filter characteristics of the matched texture filter 106.

[0042] The matched texture filter 106 (FIG. 1) of the present invention is capable of preventing the appearance of primitive edges on an image in the case of selective edge supersampling, as described in commonly-assigned U.S. Patent Application entitled "Selective Super-Sampling/Adaptive Anti-Aliasing or Complex 3D Data". For example, FIG. 6 illustrates a typical problem with internal edges. Assuming that the prescribed default mode for the matched texture filter 106 (FIG. 1) is the bilinear filtering mode and that super-sampling is performed at areas covered by pixels 602, 604, 606, 608, 610, 612 and 614, the filtering technique used at pixels 602-614 and the averaging of the subsamples (post-filtering) step for those same pixels should have a response equivalent to the default mode (i.e., bilinear filtering mode). Otherwise, a primitive edge 616 may appear as shown in FIG. 6.

[0043] Another feature in accordance with an embodiment of the present invention relates to performing matched texture filtering with a combination of linear filtering (between MIP levels) and box filtering. This embodiment provides a texture filter that behaves like a trilinear filter when, for example, the sampling rate per pixel is at the same rate as the desired screen resolution. The matched texture filter in this embodiment is capable of morphing between a bilinear filter for each MIP map level and a box and trilinear filter when the sampling rate increases per pixel (i.e., when polygon data is super-sampled). The filter characteristic is also dependent on the amount of "stretching" and "squeezing" of the texture when polygon data is super-sampled, as described above with reference to FIG. 2.

[0044] FIG. 7 illustrates the details of a matched texture filter 700 in accordance with an embodiment of the present invention. The filter 700 includes an n module 702, a δ module 704, a β module 706, an α module 708 and a blending module 710. The n module 702 determines or detects the sampling rate n for polygon data. Although the present invention describes having the n module 702 in the matched texture filter 700, alternatively, the n filter 702 may be implemented in the multi-rate data sampler 104 (FIG. 1).

12

[0045] The δ module 704 determines the degree of warping δ per texture coordinate. Alternatively, the degree of texture coordinate warping δ may be known or predetermined, which would result in the non-implementation of the δ module 704.

[0046] The β module 706 calculates the coefficient β which determines the filter shape of the matched texture filter 700. In one embodiment, the value of β depends on the sampling rate n and the degree of warping δ per texture coordinate. For example, β_u, β_v can be calculated based on equations (1) and (2). Although FIG. 7 depicts the n, δ, and β modules separately, the n, δ, and β modules may be embodied in a single filter select module (shown within dotted line 712).

[0047] The α module 708 calculates the texel blending factors α_u, α_v based upon the values of β_u, β_v in order to generate filtered texture pixel values. Further, blending module 510 performs the blending operations for the filtered texture pixel values by using blending equations. One such blending equation is:

$$C = T[\text{floor}(u), \text{floor}(v)] \bullet (1 - \alpha_u) \bullet (1 - \alpha_v) + T[\text{floor}(u) +1, \text{floor}(v)] \bullet \alpha_u (1 - \alpha_v) + T[\text{floor}(u), \text{floor}(v) +1] \bullet (1 - \alpha_u) \bullet \alpha_v + T[\text{floor}(u) +1, \text{floor}(v) +1] \bullet \alpha_u \bullet \alpha_v$$

[0048] The parameter $T[u, v]$ is the texture channel value (RGB for instance) for the point $[u, v]$, and **floor**(x) is defined as the biggest integer which is less than or equal to x; and α_u and α_v are defined as described above. C is the resulting sample value for the interpolated channel.

[0049] The C equation above illustrates channel calculation for a single texture level. In the case of MIP maps and trilinear filtering default mode, the above C equation should be done for each of two appropriate MIP levels and the two results should be blended in a conventional way.

[0050] The variation of the filter shape is preferably continuous. By making the texture filter vary continuously, artifacts normally visible during animation are not generated.

[0051] Thus, the present invention provides in various embodiments a texture filter that takes into account the sampling rate and the non-linear mapping between screen coordinates and texture coordinates. In particular, these embodiments of the

13

present invention provide a matched texture filter that continuously changes the pass band – stop band characteristics based upon the rate of sampling of polygon data and the extent of perspective warping introduced by the non-linear screen-to-texture mapping function.

[0052] It is also within the scope of the present invention to implement a program or code that can be stored in an electronically-readable medium to permit a computer to perform any of the methods described herein.

[0053] The invention has been described above with reference to specific embodiments. It will be apparent to those skilled in that art that various modifications may be made and other embodiments can be used without departing from the broader scope of the invention. Therefore, these and other variations upon the specific embodiments are intended to be covered by the present invention, which is limited only by the appended claims.